Control of the Traffic Flow in a Three-link Merge Junction via Dynamic Programming

Octavio Narváez-Aroche
ocnaar@berkeley.edu

December 8, 2016

1 The Three-link Merge Junction Problem

In this exercise we aim to minimize the Total Travel Time (TTT) of the three-link merge junction described in [1] up to T periods of time. This performance metric is

$$TTT\left(T\right)=\sum_{k=0}^{T}\sum_{i=1}^{3}x_{i}\left(k\right), \tag{1}$$

where $x_i\left(k\right)\in\mathbb{R}$ is the occupancy of link i at period k. Links 1, and 3 model a freeway. Link 2 is the onramp at the merge junction. There are $d_1\left(k\right)\in\mathbb{R}$, and $d_2\left(k\right)\in\mathbb{R}$ exogenous vehicles arriving to links 1, and 2, respectively. All vehicles flowing from link 2 join link 3 controlled with the input $u\left(k\right)\in\mathbb{R}$ using signaling devices. Defining $x\left(k\right):=\left[\begin{array}{ccc} x_1\left(k\right); & x_2\left(k\right); & x_3\left(k\right)\end{array}\right]\in\mathbb{R}^3$, the dynamics of the system can be compactly written as

$$x\left(k+1\right)=f\left(x\left(k\right),d_1\left(k\right),d_2\left(k\right),u\left(k\right)\right). \tag{2}$$

The parameters of the merge junction are the capacity of the links $c=40\,$[vehicles/period], the free-flow speed $v=0.5\,$[links/period], the congestion-wave speed $w=0.5/3\,$[links/period], the jam occupancy of the links $\bar{X}=320\,$[vehicles], the fraction of vehicles exiting link 1 that flow into link 3 $\beta=0.75$ (the remaining $1-\beta$ fraction are assumed to exit the freeway through an unmodeled exit link), the weight that determines the supply of link 3 available to link 1 $\alpha=1$, and the weight that determines the supply of link 3 available to link 2 $\bar{\alpha}=5$. The links have a length of 1[mile], and the time step is 30 [s].

2 Dynamic Programming

To find the closed-loop optimal sequence of inputs $\mu_{T-1}^{*}\left(x\left(0\right)\right):=\left\{u_0^{*}\left(x\left(0\right)\right),u_1^{*}\left(x\left(1\right)\right),\ldots,u_k^{*}\left(x\left(k\right)\right),\ldots,u_{T-1}^{*}\left(x\left(T-1\right)\right)\right\}$ that minimizes (1) using dynamic programming, we reformulate the problem in Section 1 as an optimization of the expected value over the sum of stage costs

$$E\left\{g\left(x\left(T\right)\right)+\sum_{k=0}^{T-1}g\left(x\left(k\right)\right)\right\},$$

where $g\left(x\left(k\right)\right):=\sum_{i=1}^{3}x_i\left(k\right)$ for $k=0,1,\ldots,T$. Even though the dynamic equations in (2) are deterministic, we introduce stochasticity with a discretization of the continuous state space based on the Kuhn triangulation approach described in [2], and cast the optimal control problem into a finite Markov Decision Process (MDP).

The finite state space of the MDP $\Xi\subset\mathbb{R}^3$ consists of $\prod_{i=1}^{3}n_i$ vectors ξ built from the permutations of n_i uniformly spaced values for the occupancy of the links x_i chosen from intervals of interest $\underline{x}_i\leq x_i\leq\bar{x}_i$. Fixing the exogenous inputs during the T periods such that $d_1(k):=d_1$ and $d_2\left(k\right):=d_2$, the control input space of the MDP is the finite set $U:=\left\{\underline{u},\underline{u}+\delta_u,\underline{u}+2\delta_u,\ldots,\underline{u}+\left(n_u-2\right)\delta_u,\bar{u}\right\}$, with $\delta_u:=\frac{\bar{u}-\underline{u}}{n_u-1}$ being the resolution of the grid. For every pair of $\xi\in\Xi$ and $u\in U$ the dynamics in (2) will cause the system to evolve into $f\left(\xi,d_1,d_2,u\right)$, since in general this state is not in Ξ, we search for the vertices ξ_j of the simplex in the Kuhn triangulation of Ξ that contains $f\left(\xi,d_1,d_2,u\right)$, and determine the probabilities of transition $p\left(\xi_j|f\left(\xi,d_1,d_2,u\right)\right)\geq 0$ from the barycentric coordinates of $f\left(\xi,d_1,d_2,u\right)$; this procedure is described in Appendix A.

The one-step-lookahead value iteration method [3] to solve the MDP can be written as

$$
\begin{aligned}
V_T(\xi) &:= g(\xi)\ \forall \xi \in \Xi \\
V_k(\xi) &:= \min_{u_k \in U} \quad E\left\{g(\xi) + V_{k+1}\left(f\left(\xi, d_1, d_2, u_k\right)\right)\right\} \quad \forall \xi \in \Xi,\ k = T-1, T-2, \ldots, 1, 0. \\
&\qquad\quad \text{s.t.} \quad \underline{x} \le f\left(\xi, d_1, d_2, u_k\right) \le \bar{x} \\
&= \min_{u_k \in U} \quad \left[g(\xi) + \textstyle\sum_{j=0}^{3} p\left(\xi_j | f\left(\xi, d_1, d_2, u_k\right)\right) V_{k+1}\left(\xi_j\right)\right] \quad \forall \xi \in \Xi,\ k = T-1, T-2, \ldots, 1, 0. \\
&\qquad\quad \text{s.t.} \quad \xi_j \in \Xi
\end{aligned} \tag{3}
$$

Note that $\underline{x} := \left[\ \underline{x}_1; \quad \underline{x}_2; \quad \underline{x}_3\ \right]$, and $\bar{x} := \left[\ \bar{x}_1; \quad \bar{x}_2; \quad \bar{x}_3\ \right]$ set box constraints in the state space of the system. The optimal input for $\xi \in \Xi$ at period k relative to the value function $V_k(\xi)$ is

$$
u_k(\xi) := \operatorname*{argmin}_{u_k \in U} \quad \left[g(\xi) + \textstyle\sum_{j=0}^{3} p\left(\xi_j | f\left(\xi, d_1, d_2, u_k\right)\right) V_{k+1}\left(\xi_j\right)\right] \quad \forall \xi \in \Xi,\ k = T-1, T-2, \ldots, 1, 0.
$$

The closed-loop optimal input $u_k^*(x(k))$ of the metered onramp also relies on the Kuhn triangulation of the space in order to be refined (figure 1). For each measurement of the state $x(k)$ we weight the optimal inputs of its simplex vertices $u_k(\xi_j)$ by the barycentric coordinates, so that

$$
u_k^*(x(k)) := \textstyle\sum_{j=0}^{3} p\left(\xi_j | x(k)\right) u_k\left(\xi_j\right) \quad \text{for } k = 0, 1, \ldots, T-1\ .
$$

The dynamics of the system under closed-loop optimal control become

$$
x(k+1) = f\left(x(k), d_1, d_2, u_k^*(x(k))\right).
$$

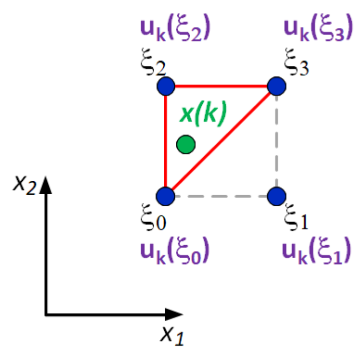


Figure 1: Refining the optimal input to continuous space in $\mathbb{R}^2$. The state feedback control input $u_k^*(x(k))$ is linearly interpolated from the optimal inputs for the vertices of the simplex $u_k(\xi_j)$ that were computed offline, using the barycentric coordinates of $x(k)$ as weights.

3 Simulation Results

To solve the three-link merge junction problem defined in Section 1 with the value iteration method in (3), we construct the finite state space $\Xi \subset \mathbb{R}^3$ by choosing $n_1 = n_2 = n_3 = 150$ for a total of $3,375,000$ MDP states within the box bounded by $\underline{x} = \left[\ 0; \quad 0; \quad 0;\ \right]$, and $\bar{x} = \left[\ \bar{X}; \quad \bar{X}; \quad \bar{X}\ \right]$; set the control space as the $n_u = 150$ evenly spread input values constrained by $\underline{u} = 0$, and $\bar{u} = c$; and consider a horizon of $T = 40$. The exogenous inputs are fixed at the limits of feasibility $d_1 = 40$, and $d_2 = 10$ reported in [1].

For the simulation of the merge junction under control of the closed-loop optimal input $\mu_{39}^*(x(0))$, we pick the uniformly spaced grid of 11^3 initial states $\underline{x} \le x(0) \le \bar{x}$ shown in figure 2. The initial states in red are described as infeasible because an optimal input $u_k^*(x(k))$ cannot be found to accommodate the exogenous incoming flows d_1, and d_2, so that for every period k, $\underline{x} \le x(k) \le \bar{x}$. For the feasible initial states, the occupancy of the links through the time horizon are shown in figures 2, and 3. The percentage of reduction in TTT achieved with the dynamic programming policy relative to the TTT of the three-link merge junction without a metered onramp (i.e., the uncontrolled traffic flow case) is in figure 4.

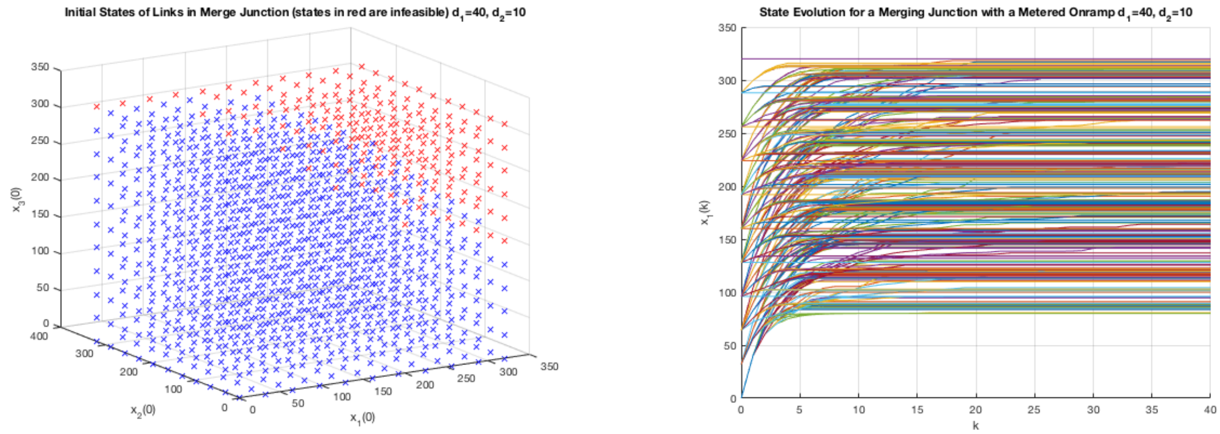


Figure 2: Grid of initial states $x(0)$ used for simulation of the three-link merge junction, the states in red are infeasible (left). Occupancy of link 1 for all feasible initial states under $\mu_{39}^{*}(x(0))$ (right).

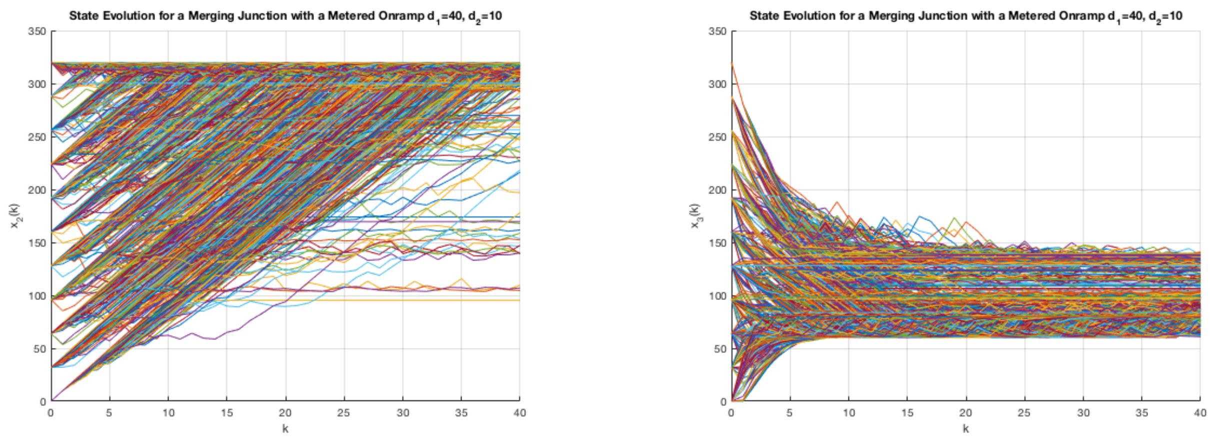


Figure 3: Occupancy of links 2, and 3 for all feasible initial states under $\mu_{39}^{*}(x(0))$.

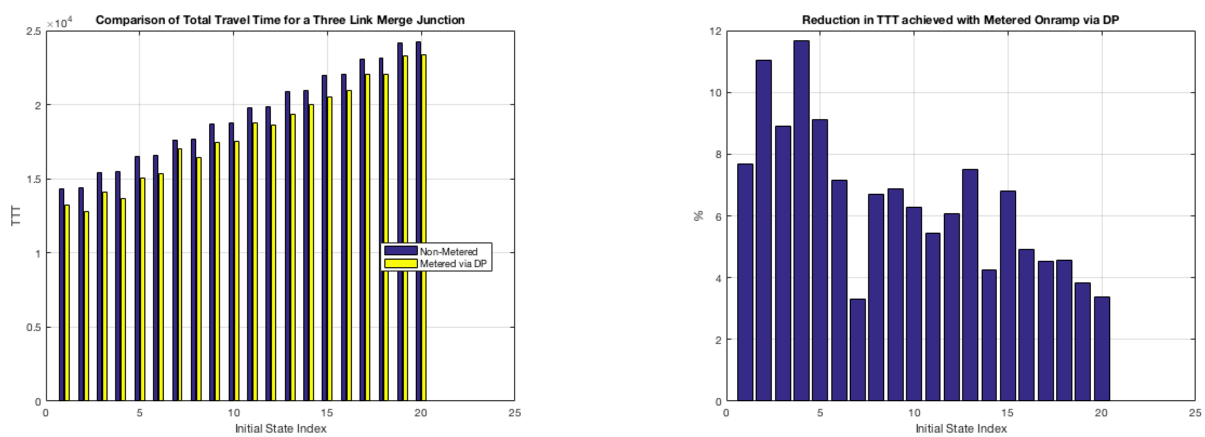


Figure 4: Comparison of the TTT for metered, and non-metered three-link merge junctions considering twenty different initial conditions. The exogenous inputs are $d_1 = 40$, and $d_2 = 10$.

4 Technical Notes

We used a workstation with four 2.7 [GHz] Intel Core i5 64 bit processors, and 8 [GB] of RAM to perform the operations described in this report. The MDP state transitions, and its corresponding probabilities from the Kuhn triangulation $p\left(\xi_j | f\left(\xi, d_1, d_2, u\right)\right)$ are computed using the Parallel Computing Toolbox in Matlab, and stored in arrays of size $n_1 n_2 n_3 \times n_u\left(n+1\right)$, where $n := 3$ is the dimension of the state space $\Xi \subset \mathbb{R}^3$. The stage cost $g\left(\xi\right) \forall \xi \in \Xi$ is stored in a $n_1 n_2 n_3 \times 1$ column vector. With this information we execute the dynamic programming algorithm to compute the value function $V_k\left(\xi\right)$, and optimal input $u_k\left(\xi\right)$, which are stored in $n_1 n_2 n_3 \times 1 \times T+1$, and $n_1 n_2 n_3 \times 1 \times T$ arrays, respectively. The optimal input array allows us to simulate the evolution of the system under the control of $\mu^*_{T-1}\left(x\left(0\right)\right)$ starting from the specified set of initial states.

Tables 1, and 2 include the memory resources, and computation time for the arrays described above, considering two different time horizons for the same set of initial states. We can see that most of the time (93.54% of total computation time for $T = 10$, and 84.67% for $T = 20$), and memory (94.46% of total memory for $T = 10$, and 89.93% for $T = 20$) are spent in acquiring the arrays for the MDP transition model, and that their sizes do not depend on the horizon T. Modifying T affects the size, and computation time for the optimal input, and value function. The simulation time is proportional to the number of feasible initial states, and the time horizon T.

Table 1: Memory, and time required for solving the three-link merge junction problem via dynamic programming with $T = 10$

Description	Type of array	Size of array	Memory size	Computation time
MDP State Transitions	uint32	$3,375,000 \times 600$	8.1 [GB]	24 [h]
MDP Probabilities of Transition[1]	uint8	$3,375,000 \times 600$	2.0 [GB]	-
Stage Cost	double	$3,375,000 \times 1$	27 [MB]	66.4 [s]
Optimal Input	double	$3,375,000 \times 1 \times 10$	270 [MB]	85.1 [min]
Value Function[2]	double	$3,375,000 \times 1 \times 11$	297 [MB]	-
Simulation[3]	-	11^3	-	13.2 [min]
Total	-	-	10.7 [GB]	25.7 [h]

Table 2: Memory, and time required for solving the three-link merge junction problem via dynamic programming with $T = 20$

Description	Type of array	Size of array	Memory size	Computation time
MDP State Transitions	uint32	$3,375,000 \times 600$	8.1 [GB]	24 [h]
MDP Probabilities of Transition[1]	uint8	$3,375,000 \times 600$	2.0 [GB]	-
Stage Cost	double	$3,375,000 \times 1$	27 [MB]	68.3 [s]
Optimal Input	double	$3,375,000 \times 1 \times 20$	540 [MB]	4 [h]
Value Function[2]	double	$3,375,000 \times 1 \times 21$	567 [MB]	-
Simulation[3]	-	11^3	-	19.6 [min]
Total	-	-	11.2 [GB]	28.3 [h]

[1] Calculated within the same function as the MDP transitions, no computation time is reported.
[2] Calculated within the same function as the Optimal Input, no computation time is reported.
[3] The value reported as "Size of array" is the number of different initial states used for simulation.

5 Brief description of Matlab functions

buildMDP Outputs the transition and probability matrices to cast the optimal control problem into a MDP.

buildMDPvaluefunctionTTT Implementation of the one-step-lookahead value iteration method to obtain the value function, and optimal inputs for the MDP states over a finite time horizon.

Demand Demand of a link for a triangular fundamental diagram in transportation networks.

GridTTT Outputs the terminal, and stage costs for the MDP states according to the Total Travel Time performance metric.

KuhnNN Determines the $n + 1$ nearest neighbors of a state in $\mathbb{R}^n$, and its barycentric coordinates relative to the Kuhn triangulation of the finite space.

MergeJunction Discrete dynamic model for the three-link merge junction.

MergeJunctionMDP Main script for solving the three-link merge junction problem via dynamic programming.

sidx2vidx Obtains the array of natural numbers *vidx* that identifies the values in the grid for each system state from the index *sidx* related to a MDP state.

state2gsidx Finds the index (a natural number, including 0) identifying the nearest MDP state to a system state x.

Supply Supply of a link for a triangular fundamental diagram in transportation networks.

vertex2sidx Provides the MDP state indices (natural numbers, including 0) identifying the vertices of a simplex in the Kuhn Triangulation of the space.

vidx2sidx Gives the MDP state index *sidx* related to the array *vidx* of natural numbers (including zero) that identify a state in the grid of the finite space.

vidx2vertex Translates an array of indices *vidx* to its corresponding MDP state. The MDP states constitute the *vertices* in the Kuhn Triangulation of the space.

References

[1] Samuel Coogan & Murat Arcak. *A Benchmark Problem in Transportation Networks.* ACM, 2016.

[2] Rémi Munos & Andrew Moore. *Variable Resolution Discretization in Optimal Control.* Kluwer Academic Publishers, Machine Learning, 49, 291-323, 2002.

[3] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, Volume 1, Third Edition, 2005.

A Kuhn Triangulation

The Kuhn Triangulation of a finite state space $\Xi \subset \mathbb{R}^n$ regards the boxes of 2^n states as composed of $n!$ simplices.

Consider a system whose dynamics are represented by $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Because in general, the input $u \in \mathbb{R}^m$ will cause the state of the system $\xi \in \mathbb{R}^n$ to evolve into a continuous state $f(\xi, u) \in \mathbb{R}^n$ outside of Ξ, we search for the vertices $\xi_{j_0}, \xi_{j_1}, \ldots, \xi_{j_n} \in \Xi$ of the simplex that contains $x := f(\xi, u)$ after a transition. The convex hull $\mathcal{CH}$ of the $n+1$ vertices is

$$\mathcal{CH} := \left\{ \lambda_{j_0}\xi_{j_0} + \lambda_{j_1}\xi_{j_1} + \ldots + \lambda_{j_n}\xi_{j_n} \mid \lambda_j \geq 0, \sum_j \lambda_j = 1 \right\}.$$

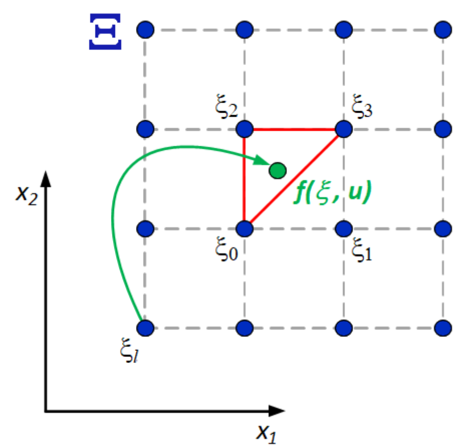
The stochastic transition model for every $\xi_l \in \Xi$ is defined as

$$p(\xi_l|x) := \begin{cases} \lambda_j, & l = j \\ 0, & \text{otherwise.} \end{cases}$$

Usually, computing the vertices ξ_j of the simplex containing x, and the coefficients λ_j are complex tasks. However, in the case of the Kuhn Triangulation [2], if the coordinates of x relative to the box where it belongs are normalized, and sorted, such that $1 \geq \bar{x}_{i_0} \geq \bar{x}_{i_1} \geq \ldots \geq \bar{x}_{i_{n-1}} \geq 0$, we have:

- The indices of the vertices $\xi_{j_0}, \xi_{j_1}, \ldots, \xi_{j_n}$ of the simplex that contains x are $j_0 = 0$, $j_1 = j_0 + 2^{i_0}$, $j_2 = j_1 + 2^{i_1}$, …, $j_n = j_{n-1} + 2^{i_{n-1}}$.
- The barycentric coordinates of x are $\lambda_{j_0} = 1 - \bar{x}_{i_0}$, $\lambda_{j_1} = \bar{x}_{i_0} - \bar{x}_{i_1}$, …, $\lambda_{j_n} = \bar{x}_{i_{n-1}} - 0$.

Figure 5 illustrates the Kuhn Triangulation of a finite state space $\Xi \subset \mathbb{R}^2$, and the process of obtaining the barycentric coordinates for a point $x \in \mathbb{R}^2$.



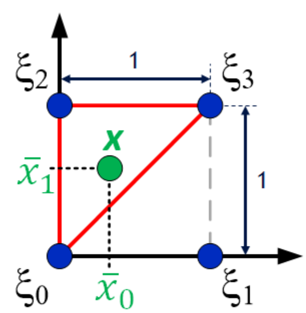


Figure 5: Kuhn Triangulation in $\mathbb{R}^2$. The boxes with 2^2 states are composed of 2! simplices, each of them with 3 vertices. The inputs cause the system to evolve from ξ to $x := f(\xi, u)$; which is in the simplex with vertices ξ_0, ξ_2 and ξ_3. The coordinates of x in the normalized box satisfy $1 \geq \bar{x}_1 \geq \bar{x}_0 \geq 0$, the indices of the vertices in the simplex that contains it are $j_0 = 0$, $j_1 = 0 + 2^1$, and $j_2 = 2 + 2^0$. The barycentric coordinates are $\lambda_0 = 1 - \bar{x}_1$, $\lambda_2 = \bar{x}_1 - \bar{x}_0$, and $\lambda_3 = \bar{x}_0$.